

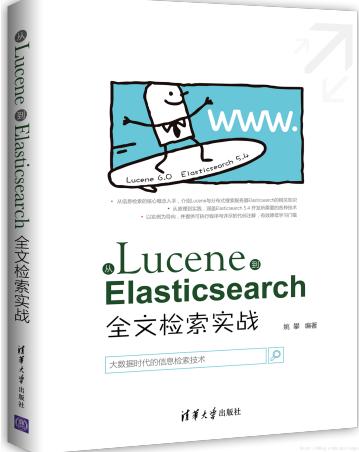
姚攀的博客 1.01^365=31.78

Engineers are versatile minds who create links between science, technology, and society.

[目录视图](#)
[摘要视图](#)
[RSS 订阅](#)

《从Lucene到Elasticsearch：全文检索实战》

希望本书能对您的工作和学习带来帮助，感谢支持！



当当 京东 亚马逊 天猫

Lucene、ES、ELK开发交流群:370734940

[加入QQ群](#)

个人资料



yyyseb

关注 发私信



访问: 843790次

积分: 8638

等级:

排名: 第2544名

图灵赠书——程序员11月书单 【思考】Python这么厉害的原因竟然是！ 感恩节赠书：《深度学习》等异步社区优秀图书和作译者评选启动！ 每周荐书：京东架构、Linux内核、Python全栈

Spring全家桶(二)Bean之间的关系、自动装配、作用域和使用外部文件

标签: [spring](#) [prototype](#) [外部文件](#)

2017-05-16 11:25

957人阅读

评论(0)

分类:

[spring](#) (8) ▾

版权声明: 本文为博主原创文章,未经博主允许禁止转载(<http://blog.csdn.net/napoay>)

[目录\(?\)](#)

[+]

五、Bean的自动装配

Spring IOC容器可以自动装配Bean，需要在bean的autowire属性里指定自动装配的模式。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:p="http://www.springframework.org/schema/p"
5   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.
6
7     <bean id="address" class="com.stuspring.autowire.Address"
8       p:city="BeiJing" p:street="HuiLongGuan">
9     </bean>
10
11    <bean id="car" class="com.stuspring.autowire.Car" p:brand="AuDi"
12      p:price="400000">
13    </bean>
14
15    <bean id="car2" class="com.stuspring.autowire.Car" p:brand="AuDi"
16      p:price="400000">
17    </bean>
18    <bean id="person1" class="com.stuspring.autowire.Person" p:name="Tom"
19      p:address-ref="address" p:car-ref="car"/>
20
21    <!--使用autowire属性指定自动装配的方式
22      byName 根据bean的id和bean的setter风格属性进行自动装配，如果有匹配的，则自动装配。
23      byType 根据类型自动装配。car和car2都满足，byType会报错。
24      确定：不够灵活，实际开发中很少使用自动装配。
25-->
```

原创: 202篇

转载: 2篇

译文: 6篇

评论: 451条

我的课程

Elasticsearch 5.4新闻搜索项目实战

StackOverFlow

<http://stackoverflow.com/users/6526424>

统计

阅读排行

- [搜索]ElasticSearch Java Api(… (61715)
- Elasticsearch索引mapping的写… (36766)
- Elasticsearch 5.X下JAVA API使… (31011)
- 搭建Elasticsearch 5.4分布式集群 (29583)
- Elasticsearch 5.1.1 head插件安… (25800)
- Elasticsearch java api(五) Bulk… (21810)
- ElasticSearch Java Api(四) -删… (19864)
- mac命令行启动tomcat (18597)
- Elasticsearch 5 Ik+pinyin分词… (17877)
- Elasticsearch shield权限管理详解 (17350)

博客专栏



Spring全家桶

文章: 10篇

阅读: 10596



MapReduce编程

文章: 7篇

阅读: 24304

Lucene&Elasticsearch&ELK
专栏

文章: 42篇

阅读: 460970



数据库

文章: 9篇

阅读: 18130



J2EE

文章: 24篇

阅读: 92700



ruby on rails

文章: 18篇

阅读: 37342

```

1 <bean id="person2" class="com.stuspring.autowire.Person" p:name="Lily"
2     autowire="byName">
3 
4 </bean>
5 
6 </beans>

```

六、 Bean之间的关系

6.1 Bean的继承关系

一个Address类，有city和Street 2个属性。

```

1 package com.stuspring.relation;
2 
3 /**
4 * Created by bee on 17/4/25.
5 */
6 public class Address {
7 
8     private String city;
9     private String street;
10    //省略setter、getter和toString方法
11 }

```

配置2个Bean:

```

1 <bean id="address" class="com.stuspring.relation.Address"
2     p:city="BeiJing" p:street="WuDaoKou"/>
3 
4 <bean id="address2" class="com.stuspring.relation.Address"
5     p:city="BeiJing" p:street="XiZhiMen"/>

```

address2和address除了street属性不一样，其他都相同，可以使用bean的集成属性来简化配置:

```

1 <bean id="address" class="com.stuspring.relation.Address"
2     p:city="BeiJing" p:street="WuDaoKou"/>
3 <bean id="address2" p:street="XiZhiMen" parent="address" />

```

1. Spring允许继承bean的配置，被继承的bean称为父bean，继承父bean的bean称为子bean。

2. 子bean从父bean中继承配置，包括bean的属性配置

3. 子bean也可以覆盖父bean继承过来的配置

4. 父bean可以作为配置模板，也可以作为bean的实例。若只想把父bean作为模板，可以设置`<bean>`的`abstract`属性为true，这样spring就不会实例化这个bean



文章: 13篇
阅读: 17547

文章分类

Elasticsearch (43)
Lucene (15)
hadoop (16)
J2EE (35)
数据结构 (16)
前端 (16)
ruby (18)
机器学习 (4)
python (5)
linux (11)
latex (2)
数据库 (12)
spring (9)
tools (1)
hibernate (1)
mybatis (1)
生活 (1)

文章存档

2017年12月 (9)
2017年11月 (1)
2017年10月 (3)
2017年09月 (2)
2017年08月 (1)

展开

最新评论

MapReduce编程(六) 从HDFS导入数据到Elasticsearch
qq_38094271 :用最新版本的那个 成功了。
注意json 文件最后一行要换行
《从Lucene到Elasticsearch:全文检索实战...
牙小木 :书已买，正在看
Lucene扩展停用词字典与自定义词库
yyseb :@cuipeng6561:这个文件在ik的目录下。
Lucene扩展停用词字典与自定义词库
cuipeng6561 :IKAnalyzer.cfg.xml 这个文件是要自己写么?
新人千万不要在 Windows 上使用 Ruby on ...
韩大嘴 :对于新手的我来说的，相当中肯！
《从Lucene到Elasticsearch:全文检索实战...
艾鹤 :6666，恭喜发财。
Elasticsearch 5.X下JAVA API使用指南
qq_41296216 :@napoay:maven坐标是 <dependency> ...
Elasticsearch java api(五) Bulk批量索引
yyseb :@wangmaohong0717:可以不指定，会自动生成。
Elasticsearch 5.X下JAVA API使用指南
yyseb :@qq_41296216:贴一下maven坐标吧，或者，博客上有QQ群，加群讨论

5. 可以忽略父bean的class属性，让子bean指定自己的类，共享相同的配置，此时abstract属性必须为true。

抽象bean例子:

```

1   <bean id="address" class="com.stuspring.autowire.Address"
2       p:city="BeiJing" abstract="true"/>
3
4   <bean id="address2" p:street="XiZhiMen" parent="address"/>
5
6   <!-- 覆盖父bean的配置-->
7   <bean id="address3" p:city="NanJing" p:street="DaFengChang"
8       parent="address"/>
```

忽略父bean的class属性:

```

1
2   <bean id="address" p:city="BeiJing" p:street="XiZhiMen" abs
3
4   <bean id="address2" class="com.stuspring.autowire.Address" parent="address"/>
5   <!-- 覆盖父bean的配置-->
6   <bean id="address3" class="com.stuspring.autowire.Address" p:city="NanJin
7       parent="address"/>
```

6.2 Bean的依赖关系

Spring允许用户通过depends-on属性设定bean前置的依赖bean，前置依赖的bean会在本bean实例化之前创建好。

如果前置依赖多个bean，可以通过逗号、空格的方式配置bean的名称。

```

1   <bean id="car" class="com.stuspring.autowire.Car" p:brand="Ford"
2       p:price="200000"/>
3
4   <bean id="person" class="com.stuspring.autowire.Person"
5       depends-on="car"/>
```

bean person依赖car， car不存在person无法实例化。

七、 Bean作用域

定义一个bean:

```
1   <bean id="car" class="com.stuspring.autowire.Car"/>
```

测试:

Elasticsearch java api(五) Bulk批量索引
hello-friend : 批量索引数据的时候，必须指定_id吗

```

1  ApplicationContext ctx=new ClassPathXmlApplicationContext
2          ("beans-scope.xml");
3  Car car1= (Car) ctx.getBean("car");
4  Car car2= (Car) ctx.getBean("car");
5  System.out.println(car1==car2);

```

打印结果：

```
1 true
```

car1和car2是同一个对象。

把作用域改为prototype:

```
1 <bean id="car" class="com.stuspring.autowire.Car" scope="prototype">
```

再次比较car1和car2，运行结果：

```
1 false
```

作用域改成prototype后，car1和car2不再是同一个对象。

在Spring中，可以在<Bean>元素的scope属性里设置Bean的作用域，默认情况下Spring只为每个在IOC容器中声明的Bean创建唯一一个实例，整个IOC容器范围内都能共享该实例。

类别	说明
singleton	在Spring的IOC容器中仅存在一个Bean实例，以单例的方式存在
prototype	每次调用getBean()都会返回一个新的实例
request	每次Http请求都会创建一个Bean,该作用域仅适用于WebApplicationContext属性
session	同一个HTTP SESSION共享一个BEAN，不同的HTTP SESSION使用不同的BEAN。该作用域仅适用于WebApplicationContext属性。

八、使用外部属性文件

配置文件里配置Bean时，有时需要在Bean的配置里混入系统部署的细节信息，比如文件路径、数据源配置信息等，这些部署细节需要和Bean配置相分离。

Spring提供了一个PropertyPlaceholderConfigurer的BeanFactory后置处理器，这个处理器允许用户将Bean配置的部分内容外移到属性文件中。在Bean配置文件里使用形式为\${var}的变量，PropertyPlaceholderConfigurer从属性文件里加载属性并使用这些属性来替换变量。

```

1 <bean id="dataSource"
2     class="com.mchange.v2.c3p0.ComboPooledDataSource">
3     <property name="user" value="root"/>
4     <property name="password" value="123456"/>
5     <property name="driverClass" value="com.mysql.jdbc.Driver" />
6

```

```

7     <property name="jdbcUrl" value="jdbc:mysql:///blog"/>
8
9 
```

测试：

```

1 package com.stuspring.properties;
2
3 import javax.sql.DataSource;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.support.ClassPathXmlApplicati
6
7 /**
8 * Created by bee on 17/4/25.
9 */
10 public class Main {
11     public static void main(String[] args) {
12         ApplicationContext ctx=new ClassPathXmlApplicationContext
13             ("beans-properties.xml");
14         DataSource dataSources= (DataSource) ctx.getBean("dataSou
15         System.out.println(dataSources);
16     }
17 } 
```

resource目录下新建db.properties,写入连接数据库的配置信息:

```

1 jdbc.username=root
2 jdbc.password=123456
3 jdbc.driverClassName=com.mysql.jdbc.Driver
4 jdbc.url=jdbc:mysql://localhost:3306/blog?useUnicode=true&characterEncoding=ut
5
6 
```

在Spring Bean的配置文件中引入context命名空间。重新配置:

```

1     <bean id="dataSource2" class="com.mchange.v2.c3p0.ComboPooledDataSource" de
2     <property name="user" value="${jdbc.username}"/>
3     <property name="password" value="${jdbc.password}"/>
4     <property name="driverClass" value="${jdbc.driverClassName}"/>
5     <property name="jdbcUrl" value="${jdbc.url}"/>
6     </bean>
7 
```

测试连接是否成功:

```

1 ApplicationContext ctx=new ClassPathXmlApplicationContext
2             ("beans-properties.xml");
3 DataSource dataSource= (DataSource) ctx.getBean("dataSource");
4 DataSource dataSource2= (DataSource) ctx.getBean("dataSource2");
5 System.out.println(dataSource);
6 System.out.println(dataSource2);
7 Connection conn=dataSource.getConnection();
8 System.out.println(conn);
9 
```

顶 踩

1 0

- 上一篇 Spring全家桶(一)HelloWorld与入门基础
- 下一篇 Spring全家桶(三)Spring SpEL

相关文章推荐

- Spring -- Bean自动装配&Bean之间关系&Bean的作用域
- MySQL在微信支付下的高可用运营--莫晓东
- Spring4.0学习笔记-bean之间的关系，作用域以及...
- 容器技术在58同城的实践--姚远
- Spring4深入理解IOC&DI02---Bean配置--自动装配...
- SDCC 2017之容器技术实战线上峰会
- spring自动装配与spring_beans之间的关系（二）
- SDCC 2017之数据库技术实战线上峰会
- Spring4学习笔记（七）：bean的作用域
- 腾讯云容器服务架构实现介绍--董明
- Bean作用域&自动装配
- 微博热点事件背后的数据库运维心得
- 不同作用域（scope）的Spring Bean之区别
- spring学习总结(六): IOC & DI 配置
- Spring4的知识应用总结（三）——Bean之间的关系
- J2EE之Spring中Bean的作用域、外部属性文件、Sp...

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

